REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* 31-08-2011	**2. REPORT TYPE** FINAL PERFORMANCE REPORT	**3. DATES COVERED** *(From - To)* JULY 2008-MAY 2011

4. TITLE AND SUBTITLE
EXPLOITING EXPLICIT AND IMPLICIT STRUCTURE IN
COMPLEX OPTIMIZATION PROBLEMS

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA9550-08-0370

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
MIFFLIN, ROBERT B.

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
WASHINGTON STATE UNIVERSITY
240 FRENCH ADMINISTRATION BLDG.
PULLMAN WA 99164-0001

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AIR FORCE OFFICE OF SCIENTIFIC RESEARCH
875 NORTH RANDOLPH ST., ROOM 3112
ARLINGTON VA 22203

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-OSR-VA-TR-2012-0273

12. DISTRIBUTION/AVAILABILITY STATEMENT
A

13. SUPPLEMENTARY NOTES

14. ABSTRACT

To solve many practical complex optimization problems often results in having to minimize a nonsmooth function f. The graph of such a function of n variables appears V-shaped in some directions while in directions orthogonal to these the graph of f's closely related \U-Lagrangian" [9] is U-shaped. To be efficient a minimization algorithm needs to approximate the correct \VU-space decomposition" and any existing second order information on the U-subspace via some kind of corresponding combination of polyhedral(V-shaped) and quadratic(U-shaped) approximation.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**	
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(Include area code)*	

Reset

Final Performance Report to Dr. Donald Hearn

Exploiting Explicit and Implicit Structure in Complex Optimization Problems

Grant Number FA9550-08-1-0370

Reporting Period: 07/01/2008 to 05/31/2011

1. Introduction

To solve many practical complex optimization problems often results in having to minimize a nonsmooth function f. The graph of such a function of n variables appears V-shaped in some directions while in directions orthogonal to these the graph of f's closely related "U-Lagrangian" [9] is U-shaped. To be efficient a minimization algorithm needs to approximate the correct "VU-space decomposition" and any existing second order information on the U-subspace via some kind of corresponding combination of polyhedral(V-shaped) and quadratic(U-shaped) approximation. For a convex function f the relevant second order information is contained in the "U-Hessian", i.e. the Hessian of its U-Lagrangian. An initial version of a VU-algorithm that does this for convex functions was published in [20]. It depends on approximating points on "primal-dual tracks" [18] where U-Hessians exist. Such approximation is based on two fundamental results depending on proximal point and bundle method theories [6]. The first is the result from [20] that says that if there exists a primal track leading to a minimizer $\bar{x}$ at which a nondegeneracy asssumption holds and a varying proximal point parameter $\mu = \mu(x)$ satisfies $\mu(x)|x - \bar{x}| \to 0$ as $x \to \bar{x}$, then for all x sufficiently close to $\bar{x}$ the proximal point of x, denoted by $p_\mu(x)$, is on the primal track. The second result from [3] implies that for any $\mu > 0$ and $x \in \Re^n$ $p_\mu(x)$ can be approximated with arbitrary precision by a finite sequence of bundle algorithm steps. In this context x is called a bundle center. Moreover, because a bundle method [6, 1] employs V-shaped (or cutting-plane) appproximation, based on gathering together a set of subgradients, it can provide local VU-decomposition basis matrices and "U-gradients" (i.e. gradients of U-Lagrangians) without having to know explicitly the underlying structure of f.

The basic VU-algorithm alternates V-steps (or corrector steps) with U-steps (or predictor steps) in an attempt to follow an unknown primal track. A U-step is a Newton step (depending on local first and second order U-derivative approximations) from a primal track point estimate p (approximating $p_\mu(x)$) to the next bundle center x_+, a point that could be a relatively poor estimate of the next primal track point. A next V-step is the final substep in a sequence of bundle algorithm substeps. It gives a correction step from x_+ to the next primal track point estimate p_+. The VU-basis matrices needed to form these steps are obtained as a by-product from computing the vector with smallest Euclidean norm in the convex hull of the subgradients active in the bundle subroutine generation of p. This short vector is denoted by s and is an estimate of a dual track point. The U-gradient required for calculation of the U-Newton step is the U-projection of s.

To prove convergence, even when a primal track does not exist, there is a line search on the line segment from p to p_+ whenever $f(p_+)$ is not sufficiently smaller than $f(p)$ relative to a multiple of $|s|^2$. This search generates a replacement for x_+ such that its corresponding p_+, coming from a second bundle subroutine run, does satisfy sufficient f-decrease.

The line search recursively builds a one-dimensional V-model of f and stops when this model is as accurate as the n-dimensional V-model generated by the bundle run that produced p. It also develops a single variable U-model as in [8]. Information gained from this search is then used in deciding a new (usually smaller) value for μ and in initializing the bundle of subgradients for the second bundle run of the current iteration.

Also, the algorithm includes another line search which may change x_+ before the first bundle run in an iteration. This extrapolation line search along the U-step direction from p is called for when a Wolfe directional derivative increase test [1] is not satisfied at x_+. This search continues until such a test is satisfied and then x_+ is redefined to be the search point found with the least f-value. Data from this search goes into possibly changing the value of μ and into initializing the bundle for the first bundle run. Moreover, at the first iteration, if an extrapolation line search is not called for, then an interpolation line search is called in order to obtain data for choosing an initial value $\mu = \mu_1$, so that the user does not have to input this value. Any line search called for by the algorithm appends up to two new subgradients to the bundle.

In addition to the above features, there is a special test for trying to detect that f is smooth in the sense that at a minimizer its U-space dimension equals n. When this f-descent test is satisfied this results in p_+ being set to x_+, i.e. there is no bundle run when the primal track is estimated to be an n-dimensional ball about a minimizer.

For various proofs, the initial version of the algorithm kept $\mu \leq \bar{\mu}$ where $\bar{\mu}$ is a fixed upper bound on the settings of μ for all bundle runs. The current version does this for all needed second bundle runs, but allows the μ-sequence corresponding to first bundle runs to go to infinity. Lemma 17 in [20], dealing with superlinear convergence, gives an implementable rate at which μ should increase when a primal-dual track is approximated sufficiently well, but for overall efficiency and convergence the algorithm was modified to include heuristics to keep μ from getting too large too soon in a run. A new proof of convergence was produced to deal with this simple modification. The observed numerical advantage for allowing μ to increase significantly at the end of a run is that the bundle subproblems do not become more difficult to solve as the iteration number increases. This is in extreme contrast to earlier proximal point methods [21] that had the parameter go to zero to obtain superlinear convergence at the cost of having the subproblems become increasingly more difficult to solve.

Some numerical reults produced by the above described algorithm are shown in Table 1 and discussed next:

	2d-U1		3d-EX		3d-U2		3d-U1		3d-U0		MAXQUAD	
	f/g	Ac	f/g	Ac	f/g	Ac	f/g	Ac	f/g	Ac	f/g	Ac
N1CV2	38	7	103	7	55	7	61	7	30	7	156	8
N.VU	12	16	21	10	12	14	13	14	18	12	50	14
qN.VU	14	12	37	11	35	11	29	15	34	14	71	11

Table 1: Summary of numerical results

The bottom three rows of the table's first column name three different methods applied to six test functions that are named in the top row of the next six columns. N1CV2 is a FORTRAN implementation of the good non-VU proximal bundle method of Lemarechal and Sagastizabal [10]. The code N.VU is written in MATLAB, except for its FORTRAN quadratic progamming subroutine taken from N1CV2. The leading N. refers to a Newton version, meaning that when it is applied to a finite max function the Hesssian of a subfunction that is active at a point is provided by the function evaluation subroutine. At each iteration a U-Hessian estimate is computed depending on a current U-projected convex combination of subfunction Hessians as described in [20]. The last method, qN.VU, is a first attempt at a quasi-Newton version of N.VU. It does not use second derivative information, but instead uses a current U-projected BFGS-updated n by n matrix intended to approximate the Hessian of the kind of Lagrangian in Theorem 7.2 of [19].

In the table, numbers under f/g indicate the total number of function evaluations for a run. Each such evaluation also includes the evaluation of one subgradient and, in the case of the Newton version, includes the evaluation of one subfunction Hessian. The label Ac indicates an accuracy measure that is the number of correct digits in the best f-value found. Except for 3d-Ex the test functions are the convex finite max functions from [20].

The convex function 3d-Ex is given by

$$f(x_1, x_2, x_3) = (1/2)x_1^2 + (1/2)\sqrt{((x_1^2 - 2x_2)^2 + (x_3 - x_2)^2)}.$$

It was created for use in conference presentations and in future publications to illustrate graphically various concepts, such as a primal and dual tracks. This nonsmooth function is the maximum eigenvalue of a symmetric 2 by 2 matrix whose three distinct entries are C^2-functions on R^3. As such, this infinite max function is in the very broad class of functions with primal-dual gradient (pdg) structure first defined in [17] and further developed in [19]. In [18] such functions are shown to have primal-dual tracks under certain conditions including "strong transversality". Also, 3d-Ex can be viewed as a quadratic function plus the Euclidean norm of a 2-dimensional vector function. N.VU was run on this function using its Hessian at points where its square root function is evaluated at a positive number and using the Hessian of the first term, $(1/2)x_1^2$, otherwise.

The good performance of the quasi-Newton version relative to that of N1CV2 as revealed in Table 1 is especially encouraging, because it is the type of algorithm which is needed to minimize implicitly defined functions resulting from applying decomposition, relaxation and/or dualization techniques to large or complex mathematical programming problems.

For the test functions with known positive definite U-Hessians the VU-codes have generated p-sequences with observed numerical superlinear convergence. For all such functions they have reported terminal V and U subspace dimensions equal to those of the ones for the known optimal solutions.

2. First year findings resulting from AFOSR support

AFOSR supported research was first devoted to working on theory for algorithm convergence and application. There are several assumptions in [20] for showing superlinear convergence of the p-sequence to a unique minimizer $\bar{x}$. One of the two most important of these is the condition that eventually each bundle run generates local V and U subspace basis matrices that converge to ones for the corresponding subspaces at $\bar{x}$. The Principal Investigator consulted with the Key Senior Investigator, Claudia Sagastizabal, on her paper with A. Daniilidis and M. Solodov [4] which shows this condition holds for certain convex max-functions. In addition to assuming that the bundle center is close enough to $\bar{x}$, the authors assume that the prox-parameter is sufficiently large, a condition that fits in well the above discussion on not having an upper bound on the μ-value chosen for the first bundle run for each iteration.

The PI corrected their initial result concerning which generated vectors are basis vectors for the V-subspace and also suggested a bundle subroutine stopping test (based on being similar to one in [20]) which turned out to be better numerically than their original one. In addition, the PI improved their discussion of the structure of the L1-norm function used as an example. This improvement, detailed next, validates the importance of pdg-structure mentioned in the introduction. The definition of pdg-structure depends on several C^2 structure functions, denoted by f_j and ϕ_j, where the latter type are essential for expressing the structure of infinite max-functions such as maximum eigenvalue functions. For $f(x) = \sum_{i=1}^{n} |x_i|$ at a point $\hat{x}$ where m out of the n $\hat{x}_i$'s are zero there are two extreme ways to express the pdg-structure for f near $\hat{x}$. One way uses 2^m structure functions f_j based on the classical result that f can be expressed as the pointwise maximum of 2^m C^2 functions . The other way developed by the PI uses only $1 + m$ structure functions, denoted by f_0 and ϕ_j for $j = 1, 2, ..., m$. An important condition that implies the existence of a primal track to $\hat{x}$ is strong transversality. For the first structure this condition says that at $\hat{x}$ the gradients of the 2^m f_j's are affinely independent. For the second structure the condition reduces to the m $\nabla\phi_j(\hat{x})$'s being linearly independent which is the case for this f, since these gradients are distinct unit vectors. If 2^m is larger than $n+1$ then the first structure does not satisfy strong transversality, so the second structure is to be preferred. In any case the second structure appears to be a more natural representation for the local structure of the L1-norm function. This is another illustration of the importance of having ϕ_j-type structure functions for describing the local structure of nonsmooth functions.

Also, during the first year of AFOSR grant research more work was done on the quasi-Newton version of the algorithm. The original version restarted a BFGS update process with a scalar multiple of the identity matrix whenever the U-dimension estimate (which starts at n) attained a new least value. The current version does not restart, but instead starts at iteration 3 or higher when two distinct p-iterates and corresponding s-iterates generate a large enough curvature estimate to be used as an initialization scalar multiplier. This simpler process runs as well as the original one on the Table 1 test functions and on several others. This change has the added benefit of allowing for fewer and less complicated heuristics for updating the prox-parameter in situations where there is some indication that its value is relatively too large, which can occur when the VU-dimensions are not yet optimal.

Another promising quasi-Newton update from [13] was also tried. However, in our reduced Hessian context it often produced ill-conditioned U-Hessian estimates which led to poor overall performance in comparison to the BFGS update. Perhaps, this indicates that in the future the DFP update should be tried, because it is one in the Broyden class that is, in some sense, furthest away from the new update in [13].

Additional testing on some randomly generated functions that are pointwise maxima of quadratics found a few functions that were much more difficult to minimize than most in terms of total number of function evaluations. An idea that led to overall improvement depends on the observation that for some bundle subproblem runs the final iterate that satisfies the stopping test has an f-value that is worse than that of one of the previous bundle iterates that was active in generating the last iterate. So, the bundle subroutine output proximal point estimate p was changed to be the bundle iterate with the best f-value from among those of either the last iterate or the iterates active in generating the last one. This change could be even more important for extending the algorithm to converge to stationary points for nonconvex functions, a subject of ongoing research.

3. Second year research findings

During the second year of research more progress was made on improving the performance of the BFGS quasi-Newton version of our VU-algorithm for the convex case. As stated in the introduction

x_+, equal to p plus its U-Newton step, may not be a "good" bundle center in the sense that its bundle run generated prox point estimate p_+ has such a large f-value that it causes both a line search and an additional bundle run to be executed, in order to obtain good replacements for the values of x_+ and p_+. In order to avoid some of these undesirable iterations having two bundle runs there is now a heuristic test based on the f-value at the initial value of x_+. One possible outcome of the test is to declare a "bad" value of x_+ and call for a preemptive line search from p along the U-Newton direction to find a replacement point that has a very good f-value, i.e. one that does not exceed $f(p)$. This action is taken when $f(x_+) > max\{f(x), f(x_-)\}$ where x is the bundle center corresponding the current p and x_- is the one from the previous iteration. Not including $f(x_-)$ in this heuristic test led to poor performance on some test functions that either did not have a primal track to a minimizer or did not have a positive definite U-Hessian at a minimizer. The current code actually has a third heuristic number to test against. Research on this important feature will be revisited when a future version of the algorithm is developed for nonconvex functions.

The quasi-Newton version of the algorithm, with the test for $f(x_+)$ being too large and with the other features mentioned above, gave very good numerical results on several test functions. These included slight improvements for the functions considered in Table 1. For example, the new numbers for the 3d-EX column are 35 14 and for the 3d-U0 column are 30 15. Also, this version provided observed superlinear convergence as illustrated below in Figure 1 where it is called VU-BFGS. This figure shows the results of running two BFGS-update based methods on an 8 variable nonsmooth example function from Lewis and Overton [11]. The function's minimum value is zero. The graph shows function value, on a log base 10 scale, depending on function evaluation number, i.e. the first blue and red marks correspond to a value around 660, the 249^{th} red dot's value is about 10^{-15} while the 87^{th} blue mark's value is near 10^{-16}. The red dots were produced by the smooth BFGS algorithm with line searches from [11] and they represent observed R-linear convergence. The blue plus symbols come from VU-BFGS and they illustrate observed R-superlinear convergence.

We are now in the position to improve the numerical behavior of the algorithm and future versions of it, even for the nonconvex case. This is because we have gotten permission to use Kiwiel's best computer code for solving the sequence of quadratic programming (QP) subproblems called for in each bundle run. This code is a very accurate one based on a specialized QP method in [7]. It also has the facility to execute a "warm start" that uses an ending matrix factorization for one QP to start the next QP solution process.

Also, during the second year of research we made two preliminary advances for getting a handle on the noncovex case. One is the improvement of a code [16] for local minimization of a nonconvex function of a single variable which can be used for executing line searches called for by

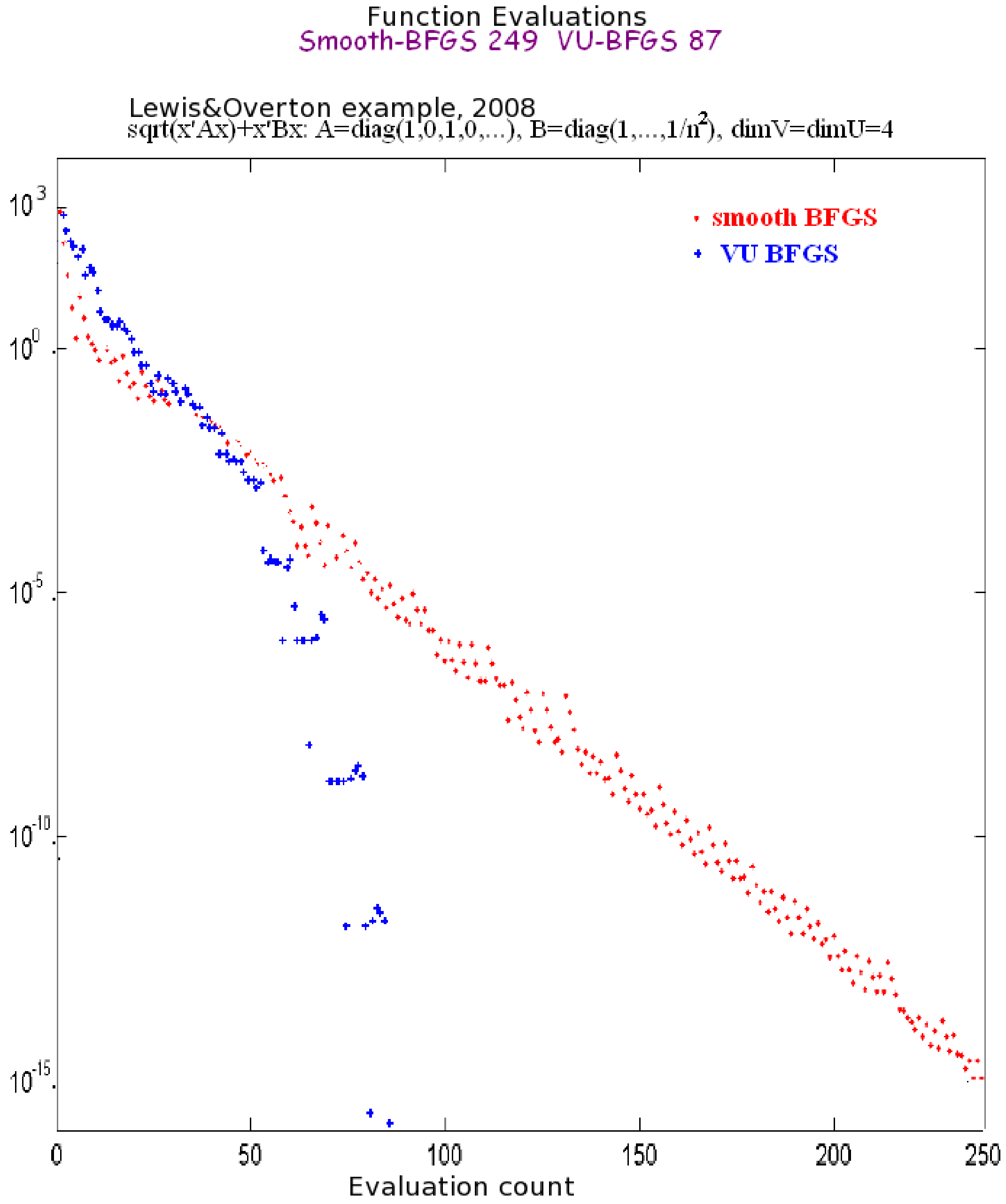

FIGURE 1. Comparison of Smooth and VU Algorithms

a multivariable algorithm and for generating useful negative curvature information when such is encountered. The code is based on the method in [15] which combines linear and/or quadratic approximation to obtain superlinear convergence for piecewise C^2 functions. The improvement comes from more use of quadratic approximation in order to increase the likelihood of an iterate being on the opposite side of a local minimizer from that of the previous iterate. This means that curvature information on either side of a local minimizer is very likely to be updated every second iteration rather than having a sequence of two or more consecutive iterations where there is no update on a given side.

The other significant advance comes from the PI's consultation with Claudia Sagastizabal on her non-VU composite bundle method [23] for solving certain nonconvex composite problems of the form

$$\min_{x \in \Re^n} (h \circ c)(x)$$

where $c : \Re^n \to \Re^m$ is a smooth mapping and $h : \Re^m \to \Re$ is a positively homogeneous convex function. Such a structure separates the two difficulties of nonconvexity and nondifferentiability by allowing only the component functions of c to be nonconvex and only h to be nonsmooth. This separation along with positive homogeneity of h allows the algorithm to only store and update a bundle of subgradients of the outer function h rather than ones from the objective $f = h \circ c$. In [23] this implementable algorithm is shown to have accumulation points that are Clarke stationary for locally Lipschitz functions. It is a specialized version of a conceptual method introduced by Lewis and Wright [12] for much more general outer functions h. These methods employ certain composite proximal linearized subproblems each of which involves replacing the inner smooth mapping c by an affine approximation. Because of this linearization, the new composite bundle algorithm has a type of step not present in previous proximal bundle methods that only have serious and null steps. This third type of step is called a backtrack step. It is like a null step in that it does not change the bundle center, but unlike one, because it requires the bundle parameter to be increased strictly. This leads to the interpretation of it being a step that decreases the size of an implicit trust region about the current bundle center.

A possible first step in a forecast future progression of steps to learn how to deal with nonconvexity within a VU-algorithm would be to determine how to use information in the composite bundle algorithm to make V-space approximations and where to add moves from corresponding U-gradient approximation.

4. Final year research findings

During this period a major revision of Sagastizabal's composite bundle method paper [23] was made and submitted to $Mathematical Programming$. It now contains a large number of computational runs on an extensive variety of convex and nonconvex test functions. This paper's algorithm is compared with three other methods, including the smooth BFGS algorithm with a Wolfe line search considered in Figure 1, and the HANSO package hybrid method downloadable from `http://cs.nyu.edu/overton/software/index.html`. This hybrid is the BFGS algorithm followed by the Gradient Sampling algorithm [2], if the BFGS terminal point is not satisfactory.

Even though BFGS does not have a convergence proof for nonsmooth functions and it has been observed to fail, it does have adequate accuracy and relatively fast convergence for some functions. The HANSO method converges in a probabilistic sense to Clarke critical points, because Gradient Sampling has this property. But this hybrid method is the most expensive one in the foursome in terms of total number of functions evaluations.

Overall, in a certain average sense including the performance measures of solution accuracy, computer time used and number of function evaluations, Composite Bundle was judged to be the best performer among the four solvers tested. Of course, a different conclusion could result from running on a different set of test functions.

Composite Bundle did have some difficulties with certain function and starting point instances. There were stalls resulting from consecutive iterations with too short function-value-reducing serious steps or with null and/or back track steps which do not improve the objective value. Such behavior results in the plateaus seen in Figure 2 below, which is taken from [24] and is discussd next.

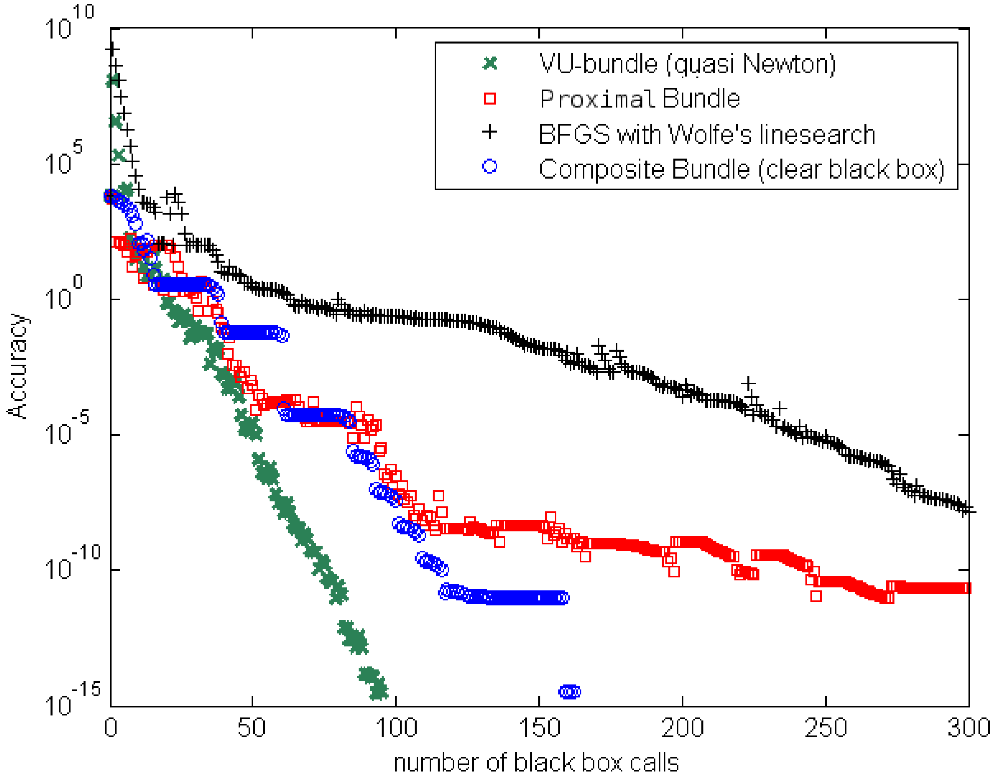

FIGURE 2. Function value sequences for `maxquad`

Figure 2 deals with the nonsmooth convex test-function called `maxquad` [1, p. 153], which also is considered above in the last column of Table 1, but with a different starting point. It is the pointwise maximum of five convex quadratic functions on $\Re^{10}$. At its unique optimal solution, four of the quadratic functions are active, but the fifth (inactive) one is stiff, making `maxquad` an interesting academic example for testing different algorithms. It is more difficult to minimize than the function considered in Figure 1.

For four different algorithms Figure 2 shows function value accuracy, relative to the smallest value ever found. The horizontal axis gives number of function evaluations referred to there as black box calls. In addition to Composite Bundle (blue circles) and smooth BFGS (black plus symbols) this algorithm group contains the Proximal Bundle[10] code N1CV2 (red squares), considered in Table 1, and VU-bundle (quasi-Newton) (green x's), called VU-BFGS in Figure 1. The VU-algorithm can be thought of as a significant improvement of both the Proximal Bundle and BFGS methods, since it combines bundled subgradients of the former, to estimate V and U subspaces, together with use of the BFGS Hessian update formula of the latter, but restricted to the U-subspace where Hessian approximation makes sense. A possible follow-up research topic would to be to make a similar improvement of Composite Bundle by adding appropriate U-steps to its serious steps. This could be an important development in nonconvex minimization, but only in the special case of known composite structure where the outer function is positively homogeneous and convex. This could provide insight into the noncovex case before attacking much more general nonconvex functions where the evaluation black box only provides one generalized gradient value with each function evaluation.

However, we speculated that the key idea of a backtrack step from the Composite Bundle method could provide enough insight to jump to the quite general case of locally Lipschitz functions that are semismooth [14]. We thought it could be the needed ingredient to make a successful VU-modification of Gupta's [5] provably convergent algorithm that bundles Hessians together with points and their associated generalized gradients (simply called gradients from here on). The bundled Hessians that give negative curvature information can be used to construct, V-models with strictly concave "sides". This can be done algebraicly by modifying bundle subproblem linearization errors via the addition of quadratic terms as done in 1980's AFOSR supported research [15] for the single variable case.

Embarking on this research fortuitously led to the most significant results we have to date for the noncovex case and bode well for proposed research under our new AFOSR grant. So far, we have discovered how to define a non-VU algorithm with finite line searches and overall convergence to local minimizers for semismooth functions without even requiring backtrack steps. The only steps needed are serious and null ones depending on newly developed Armijo and Wolfe line search tests appropriately generalized for nonsmoothness and nonconvexity. Moreover, the Hessians based on negative curvature discovered during line searches need not be stored as matrices, since they are needed only for matrix-vector products. Instead pairs of point and corresponding gradient difference vectors can be stored for use in a symmetric rank one (SR1) update formula starting from the zero matrix.

Future research will investigate if the assumption of f being lower C^2, which implies semismoothness, may be sufficient to obtain boundedness of the negative curvature matrices which is a condition needed for convergence proofs. This could be the case, because such a function has the implicit property of being the sum of a convex function and quadratic function [22]. Also, there will be work on how and when to add U-steps to the serious steps of the basic nonconvex algorithm in order to achieve rapid local convergence under appropriate assumptions.

References

[1] J.F. Bonnans, J.Ch. Gilbert, C. Lemaréchal, and C. Sagastizábal. *Numerical Optimization. Theoretical and Practical Aspects*. Universitext, 2nd edition, Springer-Verlag, Berlin, 2006.

[2] J. Burke, A. Lewis, and M. Overton, “Two numerical methods for optimizing matrix stability”, *Linear Algebra and its Applications* **351-352** (2002) 117-145.

[3] R. Correa and C. Lemarechal, “Convergence of some algorithms for convex minimization”, *Mathematical Programming* **62** (1993) 261-275.

[4] A. Daniilidis, C. Sagastizabal and M. Solodov, ”Identifying Structure of Nonsmooth Convex Functions By the Bundle Technique”, *SIAM Journal on Optimization* **20** (2009) 820-840.

[5] N. Gupta, “A higher than first order algorithm for nonsmooth constrained minimization”, Ph.D. Dissertation, Washington State University, (Pullman, WA, 1985).

[6] J.B. Hiriart-Urruty and C. Lemarechal, *Convex Analysis and Minimization Algorithms I and II* (Springer-Verlag, Berlin, 1993).

[7] K.C. Kiwiel, “A dual method for certain positive semidefinite quadratic programming problems”, *SIAM Journal on Scientific and Statistical Computing*, **10**(1) (1989) 175-186.

[8] C. Lemarechal and R. Mifflin, “A globally and superlinearly convergent algorithm for one-dimensional minimization of convex functions”, *Mathematical Programming* **24** (1982) 241-256.

[9] C. Lemarechal, F. Oustry and C. Sagastizabal, “The U-Lagrangian of a convex function”, *TAMS*, **352** (2000) 711-729.

[10] C. Lemarechal and C. Sagastizabal, “Variable metric bundle methods: from conceptual to implementable forms”, *Math. Program., Ser. A*, **76** (1997) 393-410.

[11] A.S. Lewis and M.L. Overton, “Nonsmooth Optimization via BFGS”, available at $http://www.optimization-online.org/DB_HTML/$2008/12/2172.*html*, 2008.

[12] A.S. Lewis and S.J. Wright, *A proximal method for composite minimization*, available at $http://www.optimization-online.org/DB_HTML/$2008/12/2162.*html*, 2008.

[13] C. Liu and S. Vander Wiel, ”Statistical Quasi-Newton: A New Look at Least Change”, *SIAM Journal on Optimization* **18** (2007) 1266-1285.

[14] R. Mifflin, “Semismooth and semiconvex functions in constrained optimization”, *SIAM Journal on Control and Optimization* **15** (1977) 959-972.

[15] R. Mifflin, “Stationarity and superlinear convergence of an algorithm for univariate locally Lipschitz constrained minimization”, *Mathematical Programming* **28** (1984) 50-71.

[16] R. Mifflin, “An implementation of an algorithm for univariate minimization and an application to nested optimization”, *Mathematical Programming Study* **31** (1987) 155-166.

[17] R. Mifflin and C. Sagastizabal, “On VU-theory for functions with primal-dual gradient structure”, *SIAM J. Optimization* **11** (2000) 547-571.

[18] R. Mifflin and C. Sagastizabal, “VU-Smoothness and Proximal Point Results for Some Nonconvex Functions”, *Optimization Methods and Software*, **19** (2004) 463-478.

[19] R. Mifflin and C. Sagastizabal, “Primal-dual gradient structured functions: second-order results; links to epi-derivativies and partly smooth functions”, *SIAM Journal on Optimization*, **13**(4) (2003) 1174-1194.

[20] R. Mifflin and C. Sagastizabal, “A VU-algorithm for convex minimization”, *Mathematical Programming*, **104** (2005) 583-608.

[21] R.T. Rockafellar, “Monotone operators and the proximal point algorithm”, *SIAM Journal on Control and Optimization* **14** (1976) 877-898.

[22] R.T. Rockafellar and R. Wets, **Variational Analysis**, no. **317** in Grund. der Math. Wiss, Springer-Verlag, 1998.

[23] C. Sagastizábal, “Composite Proximal Bundle Method”, available at $http : //www.optimization - online.org/DB\_HTML/2009/07/2356.html$, 2011.

[24] C. Sagastizabal, “Nonsmooth Optimization: Thinking Outside of the Black Box”, *SIAG/OPT Views-and-News*, **22** (2011) 1-9.